AD-754 376

OPTIMAL CAPACITY EXPANSION IN A FLOW NETWORK

Alan W. McMasters

Naval Postgraduate School
Monterey, California

12 September 1972

DISTRIBUTED BY:

NTIS
National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

NPS55MG72091A

United States Naval Postgraduate School

OPTIMAL CAPACITY EXPANSION IN A FLOW NETWORK

by

Alan W. McMasters

12 September 1972

Approved for public release; distribution unlimited.

36

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California	2a. REPORT SECURITY CLASSIFICATION Unclassified
	2b. GROUP

3 REPORT TITLE

Optimal Capacity Expansion in a Flow Network

4 DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Report

5 AUTHOR(S) (First name, middle initial, last name)

Alan W. McMasters

6 REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO OF REFS
12 September 1972	~~37~~ 36	12

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S) NPGS55MG72091A
b. PROJECT NO	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10 DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11 SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13 ABSTRACT

The capacity expansion problem for flow networks, first studied by D. R. Fulkerson, is reexamined. In the case where no free initial capacity is available, it is shown that the optimal expansion takes place on the arcs of the cheapest chain in the sense of unit expansion costs through the network. The proof makes use of Dantzig's decomposition principle of linear programming. In the case where some free initial capacity is available, an algorithm based on the topological dual is presented. This algorithm does not require that the flow network be planar and can be easily extended to problems having positive lower bound restrictions on arc flows, problems having bounds on individual arc expansion or nonlinear convex expansion costs, and capacity reduction problems.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

UNCLASSIFIED

Security Classification

A-31408

UNCLASSIFIED
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Network flows						
Graph theory						
Optimization						
Duality theory						

DD FORM 1 NOV 65 1473 (BACK)
S/N 0101-807-6821

UNCLASSIFIED
Security Classification

A-31409

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral M. B. Freeman, USN
Superintendent

M. U. Clauser
Provost

ABSTRACT:

The capacity expansion problem for flow networks, first studied by D. R. Fulkerson, is reexamined. In the case where no free initial capacity is available, it is shown that the optimal expansion takes place on the arcs of the cheapest chain in the sense of unit expansion costs through the network. The proof makes use of Dantzig's decomposition principle of linear programming. In the case where some free initial capacity is available, an algorithm based on the topological dual is presented. This algorithm does not require that the flow network be planar and can be easily extended to problems having positive lower bound restrictions on arc flows, problems having bounds on individual arc expansion or nonlinear convex expansion costs, and capacity reduction problems.

This research was sponsored by the Foundation Research Program of the Naval Postgraduate School.

Prepared by:

Alan W. McMasters
A. W. McMasters
Department of Operations Research
and Administrative Sciences

Approved by:

J. R. Borsting, Chairman
Department of Operations Research
and Administrative Sciences

Released by:

John M. Wozencraft
J. M. Wozencraft
Dean of Research Administration

NPS55MG72091A

12 September 1972

TABLE OF CONTENTS

1. INTRODUCTION

Suppose we have a connected network $G(N,A)$ consisting of a set of nodes N and a set of arcs A. Let the integers $i = 1,2,\ldots,P$ represent the nodes and the two-tuples (i,j) $(i = 1,2,\ldots,P;$ $j = 1,2,\ldots,P;$ and $i \neq j)$ represent the arcs. Let node 1 correspond to the source and node n correspond to the sink. The arcs are assumed directed so that the order (i,j) implies an arc directed from node i to node j.

Let $M_{ij} \geq 0$ represent the initial flow capacity of arc (i,j), $Y_{ij} \geq 0$ represent the added flow capacity, and $X_{ij} \geq 0$ represent the actual flow in the arc. Let Q represent the net flow through the network from node 1 to node $\cancel{n}$ P. Let $B \geq 0$ be the total resource budget available and let $a_{ij} \geq 0$ be the cost per additional unit of capacity added to (i,j).

The capacity expansion problem for a flow network was first studied by Fulkerson (1959). He referred to it as the parametric budget problem and stated it as follows: Find nonnegative values of X_{ij} and Y_{ij} which

$$\text{maximize } Q, \tag{1a}$$

$$\text{subject to } \sum_j (X_{1j}-X_{j1}) = Q,$$

$$\sum_j (X_{ij}-X_{ji}) = 0 \qquad (i = 2,3,\ldots,P-1), \tag{1b}$$

$$\sum_j (X_{Pj}-X_{jP}) = -Q,$$

$$X_{ij} - Y_{ij} \leq M_{ij} \qquad ((i,j) \in A), \tag{1c}$$

$$\sum \sum_A a_{ij}Y_{ij} \leq B. \tag{1d}$$

In contrast to Fulkerson we write the budget constraint (1d) in inequality form although it is obvious that equality will hold at optimality if Y_{ij} is not required to be integer valued.

An easy way to solve (1) is to construct a network having two parallel arcs in place of each original; one with a cost of zero and one with a cost of a_{ij}. The capacity of the arc having zero cost would be M_{ij} while the other would be infinite. A minor modification of the Primal-Dual algorithm of Ford and Fulkerson (1957) can then be used to find the maximal flow which can be allocated for the available budget. The resulting algorithm is, in fact, the one given by Fulkerson (1959) in his presentation of the parametric budget problem.

In this paper we would like to first consider the special form of (1) when $M_{ij} = 0$. We will state the associated optimal solution and prove it constructively using the decomposition principle for linear programs (Dantzig (1963)). We will then present an algorithm for solving the general form of (1) which makes use of the topological dual of the network.

2. OPTIMAL SOLUTIONS WHEN $M_{ij} = 0$

The form of (1c) reduces to

$$X_{ij} \leq Y_{ij}$$

when $M_{ij} = 0$ and, in fact, is equality for all arcs since the budget will be used up in obtaining the maximal possible flow. We can therefore, drop (1c) from further consideration if we rewrite (1d) as

$$\sum \sum_A a_{ij} X_{ij} = B. \tag{2}$$

The optimal solution to (1a,b) and (2) is given by the following theorem.

Theorem: The maximal flow will be sent over that chain C_m *directed from source to sink which is the shortest route when the arc length of* (i,j) *is set equal to* a_{ij}. *The value of the maximal flow and the flow capacity in each arc of that chain is given by*

$$\max Q = \frac{B}{\sum \sum_{C_m} a_{ij}} = Y_{ij}.$$

The flow capacities in all other arcs are zero.

Proof: Let us begin by adding an arc $(P,1)$ to the network and then write (1b) as $EX = 0$ where E is the node-arc incidence matrix of our augmented network, X is the augmented column vector of arc flows (with X_{P1} in the last position), and 0 is null

column vector. We write (2) as $aX \leq B$ where $a = (a_{ij})$ is a row vector with $a_{P1} = 0$ in the last position. Our problem is then to find $X \geq 0$ which

$$\text{maximize } X_{P1}, \tag{3a}$$

$$\text{subject to } EX = 0, \tag{3b}$$

$$\text{and } aX \leq B. \tag{3c}$$

In decomposed form we will combine (3a) and (3c) into the m. ter program. The flow conservation equations (3b) and $X \geq 0$ will be the constraint set of the subprogram. The master program is therefore to fin.' $\lambda_i \geq 0$, $x_s \geq 0$ which

$$\text{maximize } \sum_i bX^{(i)}\lambda_i, \tag{4a}$$

$$\text{subject to } \sum_i aX^{(i)}\lambda_i + x_s = B. \tag{4b}$$

If the network has m arcs then b is an $1 \times m + 1$ vector consisting of zeros in the first m positions and a one in the last position. The subprogram is to find $X \geq 0$ which

$$\text{maximizes } [b - \pi a]X, \tag{5a}$$

$$\text{subject to } EX = 0.$$

We are using π to represent the simplex multiplier associated with (4b). Each $X^{(i)}$ us:d in the master program is a homogeneous solu- to the subprogram and therefore we have no $\sum_i \lambda_i = 1$ constraint in the master.

The iterative solution procedure begins by finding the basic feasible solution $x_s = B$ to the first restricted master which has only one variable; namely, x_s. The multiplier π is then determined and used to provide the first objective function for the subprogram. If a solution to the subprogram can be found such that

$$[b - \pi a]X > 0 \tag{6}$$

then we use that solution, call it $X^{(1)}$, to generate a new coefficient vector for the restricted master. The second restricted master program would then be: Find $\lambda_1 \geq 0$, $x_s \geq 0$ which

$$\text{maximize } bX^{(1)}\lambda_1,$$

$$\text{subject to } aX^{(1)}\lambda_1 + x_s = B.$$

A new value of π would be determined and used to generate a new objective function for the subprogram. A new solution to the subproblem, $X^{(2)}$, would be determined and tested using (6). If it fails the test then the solution process terminates. If it passes the test then it is used to create a new restricted master program involving variables λ_1, λ_2, and x_s.

The process terminates after the k^{th} iteration where $X^{(k)}$ is the first solution from the subprogram to fail the test. The optimal solution X^* can be obtained from (7).

$$X^* = \sum_{i=1}^{k} \lambda_i X^{(i)}. \tag{7}$$

The basic feasible solution to the first restricted master was observed to be $x_s = B$. Because there is a zero coefficient for x_s in (4a) we get $\pi = 0$ as the value of the simplex multiplier for the restricted master and the objective function for the subproblem is therefore $bX = X_{P1}$. A homogeneous solution to the first subprogram is easily provided by selecting any cycle directed from the source to the sink and back to the source which includes arc (P,1) and allows as much flow as possible to pass over it. Since there are no flow capacity restrictions in the subproblem we could have $X_{P1} \to \infty$. If the solution to (3a, b, c) is unbounded this condition will be provided by $\lambda_1 \to \infty$ for some solution in (4a, b) so we need only send 1 unit over the cycle in the subproblem to indicate the homogeneous solution route. Thus, we return to the master with the vector $X^{(1)}$ consisting of plus ones corresponding to arcs on the cycle selected and zeros elsewhere. Note that $[b - \pi a]X = 1 > 0$ so (6) is satisfied.

The next restricted master program is: Find $\lambda_1 \geq 0$, $x_s \geq 0$ which

$$\text{maximizes } \lambda_1,$$

$$\text{subject to } aX^{(1)}\lambda_1 + x_s = B.$$

The optimal basic feasible solution to this problem is obviously $x_s = 0$ and

$$\lambda_1 = \frac{B}{\sum\sum_{C_1} a_{ij}},$$

where C_1 denotes the cycle just chosen in the subprogram. The associated value of π is given by (8).

$$\pi = \frac{1}{\sum\sum_{C_1} a_{ij}} . \qquad (8)$$

The objective of the subprogram is then to maximize

$$X_{P1} - \frac{aX}{\sum\sum_{C_1} a_{ij}} . \qquad (9)$$

In other words, the "cost" for sending flow over an arc in the subprogram is

$$\frac{a_{ij}}{\sum\sum_{C_1} a_{ij}} \qquad (10)$$

for all original arcs of the network and -1 for arc $(P,1)$.

The maximum value of (9) for a flow of one unit would be obtained by selecting that cycle which had the cheapest "cost" from source to sink and which returns to the source via arc $(P,1)$. Suppose we denote this cycle by C_2. The associated solution will be $X^{(2)}$ if (6) is satisfied. And (6) will be satisfied if (9) is positive. The value of (9) for this solution is

$$1 - \frac{\sum\sum_{C_2} a_{ij}}{\sum\sum_{C_1} a_{ij}} ,$$

and therefore (9) will be positive only if

$$\sum_{C_2}\sum a_{ij} < \sum_{C_1}\sum a_{ij}. \tag{11}$$

Thus we introduce a new coefficient vector into the restricted master only if the route we found on the second visit to the subprogram was shorter in the sense of the a_{ij}'s than on the first visit.

Suppose we assume that (11) is true. The new restricted master then has the form: Find $\lambda_1 \geq 0$, $\lambda_2 \geq 0$, $x_s \geq 0$ which

maximizes $\lambda_1 + \lambda_2$,

subject to $aX^{(1)}\lambda_1 + aX^{(2)}\lambda_2 + x_s = B$.

The optimal solution to this master is $\lambda_1 = x_s = 0$ and

$$\lambda_2 = \frac{B}{\sum_{C_2}\sum a_{ij}}$$

because of (11). The associated value of π is

$$\pi = \frac{1}{\sum_{C_2}\sum a_{ij}},$$

and the new subprogram objective function is

$$X_{P1} - \frac{aX}{\sum_{C_2}\sum a_{ij}}.$$

The new "cost" of sending a flow over an arc is therefore

$$\frac{a_{ij}}{\sum_{C_2}\sum a_{ij}} \tag{12}$$

for arcs of the original network.

When we compare (12) with (10) we see that only the denominators differ. Recalling that $a_{ij} \geq 0$ for all arcs we then realize that the next solution to the subprogram will be identical to $X^{(2)}$ because we will get the same cheapest route cycle. The inequality (6) will be violated and the solution process will terminate. The cycle C_2 therefore consists of the chain C_m defined in the theorem and arc $(P,1)$. The value of λ_2 is therefore

$$\lambda_2 = \frac{B}{\sum_{C_m}\sum a_{ij}} \tag{13}$$

because $a_{P1} = 0$.

The optimal solution to (3a, b, c) is $X^* = \lambda_2 X^{(2)}$ where λ_2 is given by (13). Thus the arcs on the chain having the shortest length in the sense of a_{ij}'s between the source and sink should have a flow of λ_2 and their increased arc capacities should be λ_2. The maximal flow under these conditions will be λ_2 through the network.

The following corollary is an immediate consequence of the theorem if $a_{ij} = k$ for all arcs.

Corollary: *If* $a_{ij} = k$ *for all arcs of the network then the optimal solution is to spend all of the budget on the chain having the least number of arcs between the source and the sink. If this chain has* r *arcs then*

$$\max Q = \frac{B}{rk} = Y_{ij},$$

where Y_{ij} *is the capacity of an arc on* C_m, *and all other arcs have zero capacity.*

3. FINDING AN OPTIMAL SOLUTION WHEN $M_{ij} \neq 0$

While the decomposition approach could be used to solve the problem when $M_{ij} \neq 0$, it would be very inefficient. Fulkerson's modified primal-dual algorithm is probably the most efficient. However, an interesting alternative scheme is suggested by the approaches of McMasters and Mustin (1970) and Doulliez and Rao (1971) for looking at problems involving capacity reduction and expansion. These approaches require the use of a topological dual and, as a consequence, are not particularly efficient. They do, however, have a conceptual appeal.

Both papers present solution algorithms which involve dual shortest route problems for identifying the cut sets of the primal flow problem. They are, unfortunately, restricted to planar networks. This restriction has recently been overcome by McMasters (1971) who defines a pseudo topological dual associated with a two-dimensional representation of the primal network. In that paper the special form of the dual shortest route problem is also stated and an algorithm for solving the problem is presented. This information for both undirected and directed primal networks is contained in Appendix A of this paper for the convenience of the reader.

The solution algorithm presented below begins with the construction of the topological dual or psuedo dual for the primal network. The length of the shortest and second shortest routes through the dual are then determined for dual arc lengths equal to the free capacity M_{ij} of the intersected primal arcs. The length

of the shortest route is then increased by spending some of the resource budget on that arc of the shortest route. If the entire budget can be spent without increasing the route's length to that of the second shortest route then the procedure terminates. If, on the other hand, there is still a portion of the budget left when the shortest route reaches the length of the second shortest route then the length of the third shortest route is determined and both the minimum a_{ij} arcs of the first and second shortest routes are increased until the budget is used up or the two routes attain the length of the third shortest route. The process continues in this manner until the budget is used up.

Algorithm:

1. Assign the M_{ij} values as the arc flow capacities in the flow network (primal network). Construct its topological dual using the procedures described in Appendix A. Find the shortest route through the dual and its length $L^{(1)}$. Set $n = 2$.
2. Find the n^{th} shortest loopless route through the dual network of step 1 and its length $L^{(n)}$. Appendix B contains an n^{th} shortest loopless route algorithm due to Pollack (1969) and describes the modifications needed if the primal network is nonplanar.
3. Compute

$$B^{(n-1)} = \frac{B - \sum_{k=1}^{n-1} a_k[L^{(n-1)} - L^{(k)}]}{\sum_{k=1}^{n-1} a_k}$$

where $a_k = \min a_{ij}$ associated with the arcs on the k^{th} shortest route of the dual. In the case where the $(k-1)^{th}$ and k^{th} shortest routes have a "$\min a_{ij}$" arc in common then set $a_k = 0$. If modifications were required in Pollack's algorithm because the primal network is nonplanar then any shortened arcs appearing on the k^{th} route should not be included in the a_k determination (see Appendix C).

4. Increase the lengths of the arcs associated with $a_1, a_2, \ldots, a_{n-1}$ by the amount $\Delta Y^{(n)}$ where

$$\Delta Y^{(n)} = \min\{L^{(n)} - L^{(n-1)}, B^{(n-1)}\}.$$

a) If $\Delta Y^{(n)} = B^{(n-1)}$ then terminate. The budget has been consumed and the optimal capacity increase of the primal arc associated with a_r $(r = 1,2,\ldots,n-1)$ is

$$Y_{ij}^{(r)} = \sum_{k=r}^{n} \Delta Y^{(k)}.$$

For the special case where a_r was set equal to zero in step 3 do not compute $Y_{ij}^{(r)}$. All other primal arcs will have no increase in arc capacity. The value of $\max Q$ is equal to $L^{(n-1)} + \Delta Y^{(n)}$.

b) If $\Delta Y^{(n)} = L^{(n)} - L^{(n-1)} < B^{(n-1)}$ then increase n by one and return to step 2.

4. AN EXAMPLE

Consider the flow network of Figure 1 with directed arcs and nodes numbered from 1 to 5. The source is node 1 and the sink is node 5. The numbers on each arc represent M_{ij}, a_{ij}. The value of B is assumed to be 8. The preliminary steps of the dual construction are shown on Figure 1 by the dashed lines and nodes with letter labels.

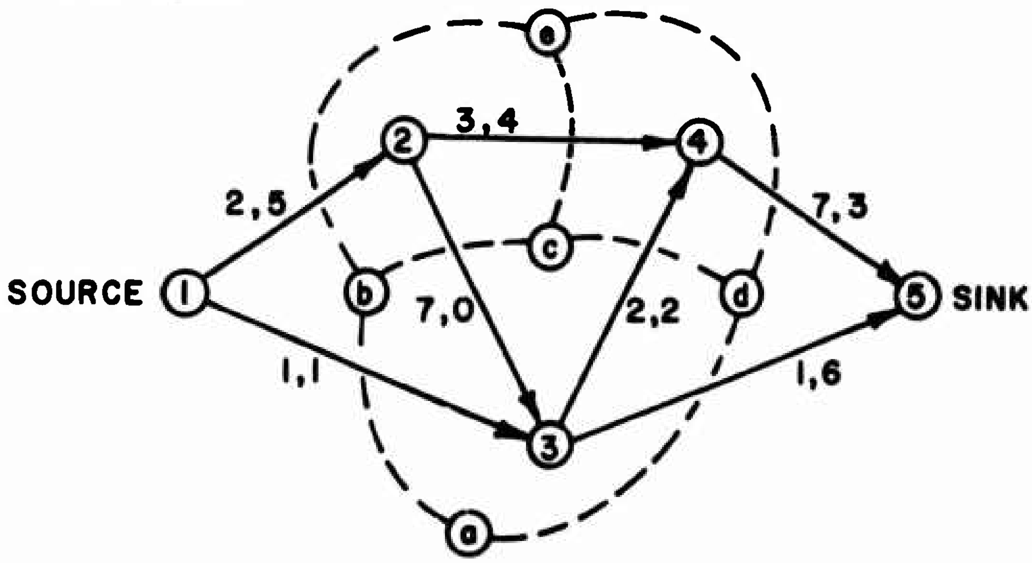

Figure 1.

The topological dual is shown in Figure 2. The numbers on each arc represent the arc's length and its associated a_{ij} value in that order. The dual origin is node a and the destination is node e.

The shortest route through the dual is a - b - e with a total length $L^{(1)} = 3$ which is the maximal feasible flow that the primal network can handle without paying for capacity. The second

shortest route without loops is a - d - c - b - e with a length $L^{(2)} = 5$. The value of a_1 is 1 and is associated with arc (a,b).

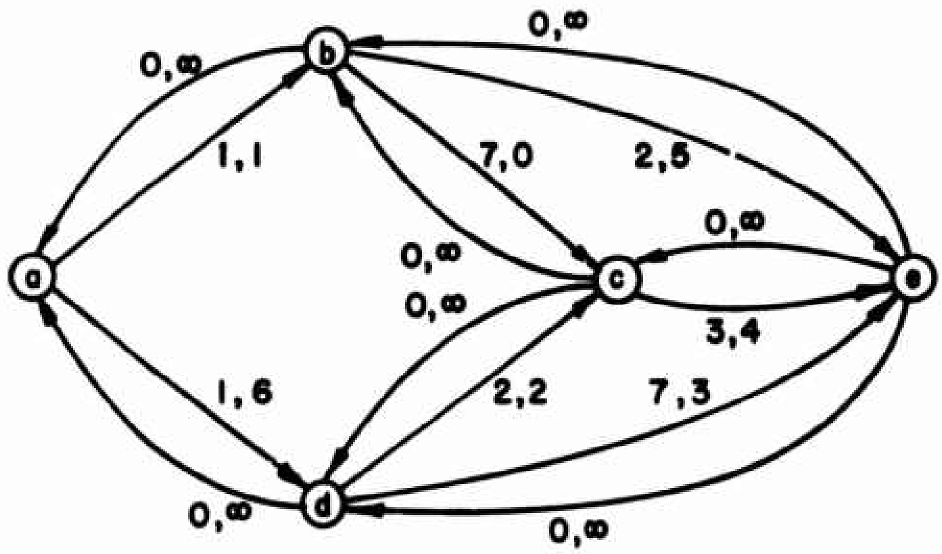

Figure 2.

Note that $B^{(1)} = B/a_1 = 8$. Because $\Delta Y^{(1)} = L^{(2)} - L^{(1)} = 2 < B^{(1)}$, two units of budget resource can be consumed before the length of the shortest route equals $L^{(2)}$.

The third shortest route is a - d - c - e with a length $L^{(3)} = 6$. The value of a_2, associated with arc (d,c), is 2 and $B^{(2)} = 2$. We lengthen the arcs associated with a_1 and a_2 by the amount $\Delta Y^{(2)} = L^{(3)} - L^{(2)} = 1$.

The fourth shortest route is a - d - e with $L^{(4)} = 8$. We get $a_3 = a_2$ because arc (d,c) has the minimum a_{ij} value on both routes and $B^{(3)} = 1$. This results in $\Delta Y^{(3)} = 1$ and the algorithm terminates.

The lengthened dual arcs are (a,b) and (d,c). Arc (a,b) has been lengthened by the amount $\Delta Y^{(1)} + \Delta Y^{(2)} + \Delta Y^{(3)} = 4$. Arc (d,c) has been lengthened by the amount $\Delta Y^{(2)} + \Delta Y^{(3)} = 2$. The primal arcs intersected by (a,b) and (d,c) are arcs (1,3) and (3,4) respectively. The capacities of these arcs have therefore been increased by 4 and 2 respectively. The maximal flow that the network can handle has been increased from 3 to 7 units.

The optimal flow through each arc which would give this maximal flow can be easily obtained by setting the arc lengths of (a,b) and (d,c) at their final values and determining the shortest distances from the origin to all nodes in the dual. It has been shown (Sakarovitch (1970)), McMasters (1971)) that the optimal flow through a primal arc in a maximal flow problem will be equal to the difference between the shortest route distances to the dual nodes incident with the intersecting dual arc. To illustrate this property we consider Figure 3 where arcs (a,b) and (d,c) have been increased in length according to the optimal solution above. The value V_i associated with each node of Figure 3 is its shortest route distance from node a. Therefore, the optimal flow through primal arc (1,3) is 5 units since $V_b - V_a = 5$. The other optimal arc flows are shown in Figure 4. The numbers on each arc correspond to $X_{ij}/(M_{ij}+Y_{ij})$.

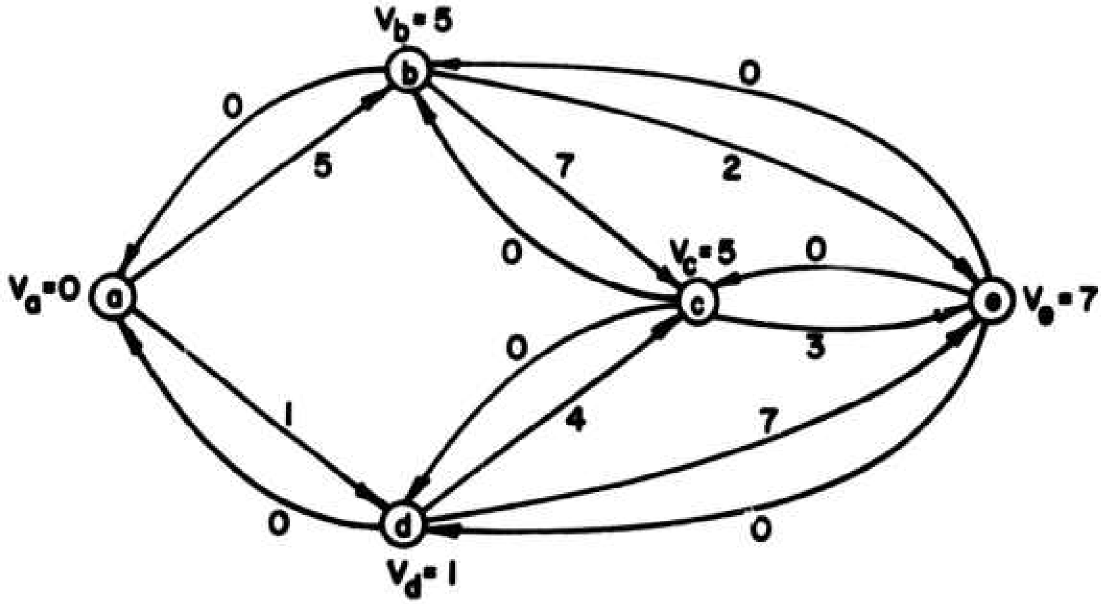

Figure 3.

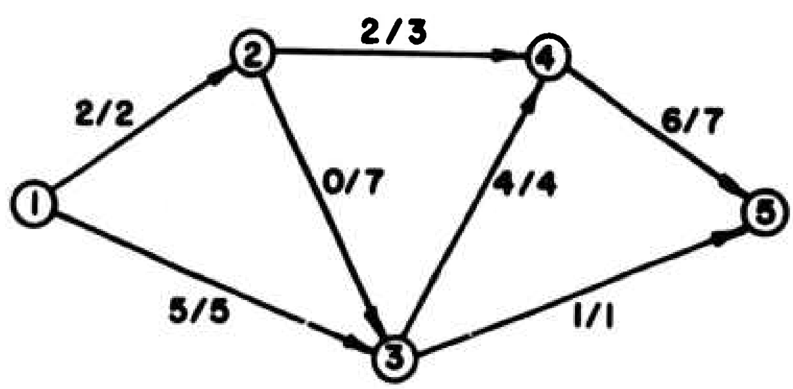

Figure 4.

5. EXTENSIONS OF THE PROBLEM

Nonzero Lower Bounds -- Consideration of nonzero lower bounds on arc flow were not explicit in Fulkerson's problem statement. Problems having negative lower bounds can, of course, be handled by replacing the primal arc by two oppositely directed arcs having zero lower bounds and positive upper bounds.

The general expression for a lower bound constraint for the parametric budget problem could be written as

$$L_{ij} \leq X_{ij} + W_{ij} \tag{17a}$$

where W_{ij} represents the amount of reduction in L_{ij}. If each unit of reduction costs b_{ij} then (1d) would take on the following form:

$$\sum \sum_A [a_{ij}Y_{ij} + b_{ij}W_{ij}] \leq B. \tag{17b}$$

The lower bound addition to the problem, regardless of the sign of L_{ij}, is easily handled by the algorithm since there is always a dual arc corresponding to the lower bound on flow (see Appendix A). In the example these arcs had zero length since $L_{ij} = 0$ for all arcs.

The b_{ij} values are assigned to the lower bound arcs in the same way as the a_{ij} values are to the upper bound arcs in the Appendix A procedure. The algorithm then considers the b_{ij}'s as merely a_{ij} values associated with certain dual arcs.

To prevent an arc associated with some L_{ij} or M_{ij} from being selected as an a_k arc, merely assign it a b_{ij} or a_{ij}

value very large. If it is not selected it will not be changed. This is, in fact, the reason for specifying infinite values of a_{ij} for the zero lower bound arcs in the procedures of Appendix A.

Nonlinear Convex $a(Y_{ij})$ -- When the cost to increase capacity $a(Y_{ij})$ is nonlinear but convex and is zero for $Y_{ij} = 0$ then a piecewise linear fit to the cost function for each arc can be made such that the algorithm can be used with minor modifications. Hu (1966), for example, suggests a linear segment for each unit of capacity change. Let $a(Y_{ij})$ represent the general form of the cost function and suppose it looks like Figure 5.

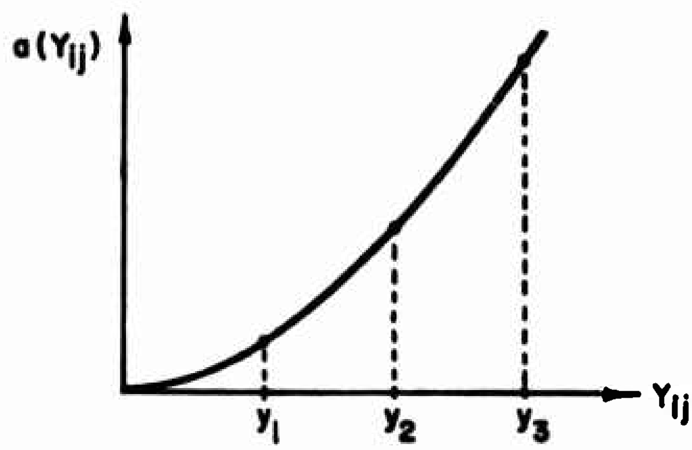

Figure 5.

The linear segment spanning $0 \le Y_{ij} \le y_1$ would have a slope

$$a_{ij}(y_1) = \frac{a(y_1)}{y_1} ,$$

the segment spanning $y_1 \le Y_{ij} \le y_2$ would have a slope

$$a_{ij}(y_2) = \frac{a(y_2) - a(y_1)}{y_2 - y_1} ,$$

and so on.

The algorithm should begin by associating $a_{ij}(y_1)$ values with each dual arc. It would proceed without change until a "min a_{ij}" arc reaches its y_1 value. If enough budget resource was available to further increase network capacity then the "min a_{ij}" arc should have its a_{ij} value changed to $a_{ij}(y_2)$ and a reappraisal of the a_{ij} values for arcs of the dual route currently under consideration should be made to see if some other arc now has a lower a_{ij} value. If so, the length of the new "min a_{ij}" arc should be increased and the first one ignored. The equation for $B^{(n-1)}$ in step 3 of the algorithm should be modified to incorporate these changes in a_k and an additional term should be added to the $Y^{(n)}$ equation in step 4 to accommodate the break points y_1, y_2, etc., in the piecewise linear fits to the $a(Y_{ij})$ curves of the dual arcs being lengthened.

These modifications would be directly applicable to problems having linear $a(Y_{ij})$ curves but upper bounds on Y_{ij} values.

An Interdiction Problem -- The interdiction problem studied by McMasters and Mustin (1970) seeks to spend money to reduce the maximum possible flow of an enemy's supply network. This is done by reducing the capacities of certain arcs of the network. If Z_{ij} is the amount of capacity reduction of arc (i,j) and a_{ij} is the unit reduction cost then (1d) would have the form

$$\sum \sum_A a_{ij} Z_{ij} \leq B.$$

If M_{ij} is the arc capacity before interdiction and $m_{ij} \geq 0$ is the least possible capacity after interdiction then (18) and (19) describe the bounds on Z_{ij} and X_{ij}.

$$0 \leq Z_{ij} \leq M_{ij} - m_{ij}. \tag{19}$$

$$0 \leq X_{ij} \leq M_{ij} - Z_{ij}. \tag{20}$$

The algorithm of section 3 is easily modified to solve this problem. The a_1 arc of the dual shortest route would be shortened in length until it reached its m_{ij} value or the budget is spent. If the m_{ij} value is reached first then shorten the arc having the next smallest a_{ij} value until it attained its m_{ij} value or the budget is spent. Repeat this process for each successive arc of the dual shortest route. The final length of the route should then be recorded.

The dual second shortest route should be analyzed in the same fashion and its final length compared with that of the shortest route. That route having the shortest final length is retained for further comparisons. This process must be repeated for each route through the dual. The shortest final route crosses the primal arcs to be interdicted. The amount of effort allocated to each primal arc depends on the length of the intersecting dual arc.

The algorithm presented in the McMasters and Mustin paper has this same flavor but is more efficient if $m_{ij} > 0$ for some arcs. Their procedure begins by finding the shortest route through the dual when all arcs of the dual are set at their m_{ij} values. The amount of money needed to attain the shortest route length is then computed. If it exceeds the budget then some "unspending" is required. If it does not exceed the budget the problem is solved.

The "unspending" looks for the most expensive arc (that corresponding to $\max a_{ij}$) on the dual shortest route. The length of that arc is then increased until it reaches its M_{ij} value or the budget constraint is satisfied. If it reaches its M_{ij} value first then the next most expensive arc is also lengthened. The process repeats until the budget constraint is satisfied. The final length of the shortest route is recorded.

The second shortest dual route based on m_{ij} values is next determined and its length checked against the final length of the shortest route. If it is longer than that final length the problem is solved; otherwise one or more arcs of the second shortest route are lengthened to meet the budget. The final length of the second shortest route is compared with that of the shortest and the route with minimum length is retained for further comparisons. The process continues until all dual routes have been examined or some dual route having all arcs at their m_{ij} values exceeds in length the shortest preceding route meeting the budget constraint.

6. APPENDIX A

Construction of the Topological Dual -- The original flow network will be called the primal network. A mesh of a planar primal network is any region surrounded by nodes and arcs but containing neither in the plane on which the network is constructed. The region of the plane completely surrounding the primal network will be called the external mesh. The construction of the dual of a source-sink planar directed network consists of the following steps (McMasters (1971)).

1. Denote the original maximal flow network as the primal network. Connect an artificial arc between the sink and source of the primal and position it below the network. The resulting network will be referred to as the modified primal network.
2. Place a node in each mesh of the modified primal including the external mesh. Let the origin of the dual be the node in the mesh involving the artificial arc and the destination be the node in the external mesh.
3. For each arc in the primal (except the artificial arc) construct two oppositely directed arcs that intersects it and join with nodes in the meshes adjacent to it.
4. Assign the value of the upper bound capacity of the primal arc as the length of the intersecting dual arc having the same direction that the primal arc would have if it were rotated 90° counterclockwise. Assign to the oppositely directed dual arc a length equal to the negative of the lower bound capacity of the primal arc (these lengths will all be zero for problem (1)).

If the primal network is not source-sink planar then modify step 1 above to read:

1. Construct a two-dimensional representation of the flow network such that all arcs are straight lines. Assign a psuedo node to every intersection of arcs not at a node in this representation. Connect an artificial arc between the sink and source of this representation and position it below the network. The resulting network will be referred to as the modified primal network.

The remainder of the steps for dual construction are the same as above.

<u>Assignment of</u> a_{ij} <u>Value</u> -- Assign the a_{ij} values of the original primal network to those dual arcs having lengths equal to the M_{ij} values. Assign $a_{ij} = \infty$ to all dual arcs corresponding to the lower bound capacity of the primal arc.

7. APPENDIX B

<u>An n^{th} Shortest Loopless Route Algorithm</u> -- We are interested in loopless routes (contain no cycles) through the dual because we want those routes corresponding to primal cut sets[1] which disconnect the primal into two subgraphs, one containing the source and the other the sink. We know that any route containing a cycle would not correspond to such a cut set (Ford and Fulkerson (1956)). We are also interested in having an algorithm which allows us to get the next shortest route at any time. Pollack (1969) developed the following algorithm which has both of these features.

1. Determine the shortest route through the network using an algorithm such as Dijkstra's if all arc lengths are non-negative or Yen's if some arc lengths are negative (see Dreyfus (1969)).[2]
2. To determine the second shortest route remove the first arc from the shortest route and solve for the shortest route through the remaining network. This route is a candidate for the second shortest route. Record the route and its length and replace the arc. Remove the second arc of the shortest route, solve for the shortest route through the remaining network, record its length and replace the second arc. Continue this process until all arcs

---

[1] A cut set of a connected graph is defined to be a disconnecting set of arcs which contains no proper subset which also disconnects the graph.

[2] Negative arc lengths will occur in the dual networks of problems having positive lower bounds on arc flows. An infeasible flow is detected by a cycle of negative length in the dual (McMasters (1971)).

on the shortest route have been removed and replaced. Examine the list of second shortest route candidates. That candidate having the shortest length is the second shortest route. In the case of a tie between two candidates arbitrarily select one as the second shortest and specify the other as the third shortest.

3. If there were no ties for the second shortest route, the third shortest route candidates are determined by removing an arc from the shortest and an arc from the second shortest routes and solving for the shortest route through the remaining network. This must be done for all combinations of two arcs from the shortest and second shortest routes. The list of candidates is then examined and the third shortest route is that candidate having the shortest length.
4. The n^{th} shortest route is obtained by first finding the shortest route through the remaining network after each combination of arcs, one from each of the preceding n-1 shortest routes, has been removed. That candidate having the shortest length is the n^{th} shortest route.

Modifications for a Nonplanar Primal -- If the primal network is not source-sink planar then loopless routes through the dual may not correspond to cut sets of the original primal. Modifications to Pollack's algorithm may therefore be required to insure selection of the correct dual routes.

Let the dual nodes be numbered $1,2,\ldots,N$ where 1 corresponds to the dual origin and N corresponds to the dual destination or terminal node. Let the group of four dual nodes surrounding a primal pseudo node ϕ be represented by $(i,j,k,m)_\phi$. Let Φ be the set of all such groups in a dual network. Let V_i represent the permanent label on dual node i after the shortest route from the dual origin $(V_1 = 0)$ to every dual node has been determined. Let ℓ_{ij} be the length of the dual arc (i,j). If, during each step of Pollack's algorithm,

$$V_j - V_i = V_m - V_k \tag{18}$$

for all node sets of Φ then no modifications are necessary. If (18) is not satisfied for some set then use the following subroutine to change the V_i values. McMasters (1971) explains the reasoning behind condition (18) and proves that this subroutine does accomplish the desired result.

Adjustment Subroutine:

1. Compute a set of numbers Y_i $(i = 1,2,\ldots,N)$, where $Y_N = V_N$ and

$$Y_k = \max_j (Y_i - \ell_{ij})$$

for $i = 1,2,\ldots,N-1$.

2. For each set $(i,j,k,m)_\phi$ for which (18) is not satisfied compute the following numbers:

$$\ell_1(\phi) = \min\{\ell_{ij}, V_j - Y_i, V_m - Y_k\}$$

$$\ell_2(\phi) = \min\{-\ell_{ji}, Y_j - V_i, Y_m - V_k\}$$

If $\ell_1(\phi) < \ell_2(\phi)$ for one or more sets then go to step 3; otherwise, go to step 4.

3. Determine that set $(i,j,k,m)_{\hat{\phi}}$ corresponding to

$$\ell_2(\hat{\phi}) - \ell_1(\hat{\phi}) = \max_{\phi}\{\ell_2(\phi) - \ell_1(\phi) \mid \ell_1(\phi) < \ell_2(\phi)\}$$

and change ℓ_{ij}, ℓ_{ji}, ℓ_{km}, and ℓ_{mk} of the arcs associated with that set to

$$\ell'_{ij} = \ell'_{km} = \ell_1(\hat{\phi})$$

$$\ell'_{ji} = \ell'_{mk} = -\ell_1(\hat{\phi}).$$

Go to step 5.

4. Select any set $(i,j,k,m)_\phi$ for which (18) is not satisfied and change the values of ℓ_{ij}, ℓ_{ji}, ℓ_{km}, and ℓ_{mk} on the arcs associated with that set to

$$\ell'_{ij} = \ell'_{km} = \ell_1(\phi),$$

$$\ell'_{ji} = \ell'_{mk} = -\ell_1(\phi).$$

5. Recompute the shortest routes from the dual origin to each node.
   (a) If (18) is not satisfied, return to step 1 of this subroutine.
   (b) If (18) is satisfied, record the shortest route through the dual and its length. Identify any shortened arcs on this route. If this route is later found to be the k^{th} shortest route then these arcs should not be "removed" on subsequent iterations of Pollack's alforithm. Return all shortened arcs to their original lengths and continue with Pollack's algorithm.

The adjustment subroutine artificially reduces the lengths of certain arcs of the dual. At the end of the subroutine one or more of these arcs may appear on the shortest route. In searching for the next shortest route using Pollack's algorithm, these shortened arcs should not be removed since "removal" is normally accomplished in shortest route algorithms by assigning an infinite length to the arc. Increasing the original length of a shortened arc would result in the route just determined by the adjustment subroutine being also determined as a candidate for the next shortest route.

8. APPENDIX C

Selection of a_k Arcs for a Nonplanar Primal -- The adjustment subroutine of Appendix B may cause arcs on a dual k^{th} shortest route to be shortened. Because increasing the original length of a shortened arc would not increase the length of the k^{th} shortest route we would gain no increased primal flow capacity. Therefore, we should select that arc corresponding to "min a_{ij}" from only those arcs on the route which were not shortened by the subroutine.

9. REFERENCES

[1] Clarke, S. A., A. Krikorian and J. Rausen, "Computing the N Best Loopless Paths in a Network," *J. SIAM*, Vol. 11, No. 4, 1963.

[2] Dantzig, G. B., *Linear Programming and Extensions*, Princeton University Press, 1963.

[3] Doulliez, P. J. and M. R. Rao, "Capacity of a Network with Increasing Demands and Arcs Subject to Failure," *Operations Research*, Vol. 19, No. 4, July-August 1971.

[4] Dreyfus, S. E., "An Appraisal of Some Shortest Path Algorithms," *Operations Research*, Vol. 17, No. 3, May-June 1969.

[5] Ford, L. R. and D. R. Fulkerson, "Maximal Flow Through a Network," *Canad. Jour. Math.*, Vol. 8, No. 3, 1956.

[6] Ford, L. R. and D. R. Fulkerson, "A Primal-Dual Algorithm for the Capacitated Hitchcock Problem," *NRLQ*, Vol. 4, No. 1, March 1957.

[7] Fulkerson, D. R., "Increasing the Capacity of a Network: The Parametric Budget Problem," *Management Science*, Vol. 5, No. 4, July 1959.

[8] Hu, T. C., "Minimum-Cost Flows in Convex-Cost Networks," *NRLQ*, Vol. 13, No. 1, March 1966.

[9] McMasters, A. W. and T. M. Mustin, "Optimal Interdiction of a Supply Network," *NRLQ*, Vol. 17, No. 3, September 1970.

[10] McMasters, A. W., "Maximal Flow Capacity of a General Network," Tech. Report NPS55MG71121A, December 1971.

[11] Pollack, M., "Shortest Route Solutions of the K-th Best Route Problem," Paper R-WA1.7, 26th Nat'l ORSA Meeting, Miami Beach, Florida, November 1969.

[12] Sakarovitch, M. F., "A Note on Duality in Network Flows," Report ORC 70-36, Oper. Res. Center, University of California, Berkeley, November 1970.